

Assignment 10-3 Identify identifiers of C++ programs 3

An identifier is a series of characters consisting of letters, digits and underscores (`_`) that does not begin with a digit. You are given a syntactically correct C++ program, and you'll have to find out all non-keyword identifiers.

Input

The input consists of exactly one line with a file name. The corresponding file which contains a syntactically correct C++ program. For the sake of simplicity, we suppose that, in this cpp file, all comments are *single-line comments* which begin with `//`.

Output

You are to output a text file which consists of all non-keyword identifiers in the inputted cpp file.

Sample Input (the content of a cpp file)

test1.cpp

The contents of the file `test1.cpp` is as follows:

[illegible]

```

        "static", "struct", "switch", "typedef",
        "union", "unsigned", "void", "volatile",
        "while", "bool", "catch", "class",
        "const_cast", "delete", "dynamic_cast",
        "explicit", "false", "friend", "inline",
        "mutable", "namespace", "new", "operator",
        "private", "protected", "public",
        "reinterpret_cast", "static_cast", "template",
        "this", "throw", "true", "try", "typeid",
        "typename", "using", "virtual", "include" };

int main()
{
    vector< char * > program;

    // reads in a C++ program from a cpp file, and put it to the vector program
    load( program );

    vector< char * > identifiers;
    for( size_t i = 0; i < program.size(); i++ )
    {
        delComment( program[ i ] ); // deletes the comment beginning with "//"
        delStrConsts( program[ i ] ); // deletes all string constants from
        delCharConsts( program[ i ] ); // deletes all character constants from

        if( strcmp( program[ i ], "" ) != 0 )
            extractIdentifiers( program[ i ], identifiers );
        // extracts all identifiers from program[ i ], and put them into the
    }

    // stores all non-keyword strings in the vector identifiers into a text
    store( identifiers );

    for( size_t i = 0; i < identifiers.size(); i++ )
        delete[] identifiers[ i ];

    for( size_t i = 0; i < program.size(); i++ )
        delete[] program[ i ];
}

void load( vector< char * > &program )
{
}

}

void delComment( char *sourceLine )
{
    size_t length = strlen( sourceLine );
    if( length > 1 )
        for( size_t i = 0; i < length - 1; i++ )
            if( sourceLine[ i ] == '/' && sourceLine[ i + 1 ] == '/' )
            {
                sourceLine[ i ] = '\0';
                return;
            }
}

void delStrConsts( char *sourceLine )
{
}

}

```

```

void delCharConsts( char *sourceLine )
{

}

void extractIdentifiers( char *sourceLine, vector< char * > &identifiers )
{

}

void store( vector< char * > &identifiers )
{

}

bool keyword( char str[] )
{
    size_t numKeywords = sizeof( keywords ) / 20;
    for( size_t i = 0; i < numKeywords; i++ )
        if( strcmp( keywords[ i ], str ) == 0 )
            return true;
    return false;
}

bool duplicate( vector< char * > &identifiers, int pos )
{
    for( int i = 0; i < pos; i++ )
        if( strcmp( identifiers[ i ], identifiers[ pos ] ) == 0 )
            return true;
    return false;
}

```